Continuous Summarization of Streaming Spatio-Textual Posts

Dimitris Sacharidis
Technische Universität Wien
dimitris@ec.tuwien.ac.at

Paras Mehta
Freie Universität Berlin
paras.mehta@fu-berlin.de

Dimitrios Skoutas
IMIS, Athena R.C.
dskoutas@imis.athena-innovation.gr

Kostas Patroumpas
IMIS, Athena R.C.
kpatro@imis.athena-innovation.gr

Agnès Voisard
Freie Universität Berlin
agnes.voisard@fu-berlin.de

ABSTRACT

In this paper, we address the problem of continuously maintaining a concise, diversified summary of the contents of a sliding window over a stream of geotagged posts. Selecting posts to include in the summary takes into account both the criteria of coverage and diversity, and the summary is updated dynamically when the window slides. Our proposed strategy provides a trade-off between information quality and performance. An experimental evaluation of our method is presented using two real-world datasets containing spatio-textual posts from Twitter and Flickr.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; **Data streaming**; **Information retrieval diversity**;

KEYWORDS

spatio-textual streams, continuous diversification, summarization

1 INTRODUCTION

A large number of spatio-textual posts, such as geotagged tweets and photos, are generated constantly by users in social media. Combining textual content with geospatial information provides a valuable source for analysis, e.g., for identifying and monitoring trending events or topics at various locations, studying the spatial distribution of opinions and sentiments associated with various entities. However, given the high rate at which this content is produced, it can easily become overwhelming for the user to keep track of the whole stream of information. Moreover, there is typically a high degree of repetition and redundancy in the contents of the streams. Thus, it is often desirable to compute and maintain a more concise, aggregate *summary* of relatively few, representative posts.

The seminal work of [2] studied the problem of document summarization and formulated it as an instance of the diversification problem, which has been extensively studied in the fields of information retrieval and web search [3, 5, 11]. Constructing a document summary that is more diverse can increase the coverage of different topics, aspects, opinions or sentiments, thus reducing repetition and bias. Several formulations exist for defining the objective of search results diversification [3, 5, 11]. According to the most well-known [5], given a set of relevant documents, the goal is to select a much smaller subset that maximizes an objective function, which combines: (a) a *relevance score*, assessing how relevant each document is, and (b) a *diversity score*, measuring how diverse the documents in the subset are. Under these formulations, finding the exact result set that maximizes the diversification objective is NP-hard. Thus, approximate solutions are proposed, relying either on *greedy* heuristics, which build the diversified set incrementally, or on *interchange* heuristics, which gradually improve upon a randomly selected initial set by swapping its elements with other ones that improve its diversity.

Most of the existing approaches address *static* settings, with very few considering a *streaming* context. In [4], continuous diversification under a sliding window model is studied. The algorithm, however, is impractical for massive, frequently updated streaming data, as it needs to maintain a cover tree storing *all* posts in the current window. The work in [9] assumes a *landmark window* model, i.e., a window over the stream that spans from a fixed point in the past until the present. The online algorithm checks every new incoming post against those in the current result set, and performs a substitution if it increases the objective score of the set.

In this paper, we focus on computing and maintaining spatially and textually diversified summaries over a stream of spatio-textual posts. We adopt the *sliding window* model by examining successive chunks of the incoming stream and incrementally updating the resulting summary to reflect recently trending posts. We formally define the problem of summarizing a stream of spatio-textual posts over a sliding window, defining specific spatio-textual criteria of coverage and diversity. We observe that our formulation is related to the max-sum diversification problem, and thus we adapt the method from [9] so as to operate under the sliding window model, a task which requires handling post expirations. Finally, we present an experimental evaluation of our approach using real-world datasets from Twitter and Flickr.

2 SUMMARY DEFINITION

Our goal is to continuously maintain a summary of the contents of a *sliding window* over a stream of *spatio-textual posts*. We formally define these concepts below.

, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Definition 1. *A* spatio-textual post $p = \langle \Psi, \ell, t \rangle$ *consists of a set of keywords* Ψ *from a vocabulary* V*, and was generated at location* ℓ *(a pair of coordinates* (x, y)*) at timestamp* t*.* □

Definition 2. *A* time-based sliding window $\mathcal{W}$ *comprises: (a) a* range *spanning over the most recent* ω *timestamps backwards from current time* t_c*, and (b) a* slide step *of* β *timestamps. Upon each slide,* $\mathcal{W}$ *moves forward and provides all messages posted during time interval* $(t_c - \omega, t_c]$*. These messages comprise the current* state *of the window, i.e.,*

$$\mathcal{W} = \{p : p.t \in (t_c - \omega, t_c]\} \tag{1}$$

Posts with timestamps earlier than $t_c - \omega$ *are called* expired. □

Given a constraint on the maximum summary size, our objective is to construct a summary that *covers* as much as possible the entire set of posts in the current window while at the same time containing *diverse* information as much as possible. Formally, we capture these two requirements using the two measures defined next.

Definition 3. *The* coverage $cov(S)$ *of a summary* S *captures the degree to which the posts in the summary approximate the spatial and textual information in the window. We use a weight* α *to capture the relative importance of the two information facets:*

$$cov(S) = \alpha \cdot cov^T(S) + (1 - \alpha) \cdot cov^S(S). \tag{2}$$

□

Similar to [7], we define the *textual coverage* of a summary as

$$cov^T(S) = \sum_{p_i \in \mathcal{W}} \sum_{p_j \in S} sim^T(p_i, p_j) \tag{3}$$

where $sim^T(\cdot, \cdot)$ is a textual similarity metric between posts.

For our purposes, we consider the vector space model, and define $sim(\cdot, \cdot)$ as the *cosine similarity* of the vector representations of the posts. Specifically, each space coordinate corresponds to a keyword, and the vector's coordinate contains a weight representing the importance of the corresponding keyword relative to the window. While any tf-idf weighing scheme [8, 10] is possible, here we simply use term frequency and normalize the vectors to unit norm. Therefore, the textual coverage is computed as the sum over each pair of posts (one from the window and another from the summary) of the inner product of their vector representations:

$$cov^T(S) = \sum_{p_i \in \mathcal{W}} \sum_{p_j \in S} \sum_{\psi} p_i[\psi] \cdot p_j[\psi], \tag{4}$$

where keyword ψ is used to index the vector and thus $p[\psi]$ denotes the normalized weight of keyword ψ of post p.

For the *spatial coverage*, we follow a similar formulation and define it as cosine similarity in a (different) vector space. Instead of keywords from a vocabulary, we have a set of regions from a predetermined spatial partitioning ρ (e.g., regions could represent cells of a uniform grid). Intuitively, such a coarse partitioning allows for a macroscopic view of the posts in the window, where exact post locations are not important and thus coalesced into broader regions. As each post is always associated with a single region, the spatial content of a post is simply represented as a vector having a single weight 1 at the vector coordinate representing the region containing the post's geotag. Thus, the spatial coverage is computed as:

$$cov^S(S) = \sum_{p_i \in \mathcal{W}} \sum_{p_j \in S} \left| \rho(p_i.\ell) = \rho(p_j.\ell) \right|, \tag{5}$$

where $\rho(\ell)$ is the region associated with location ℓ, and $|\rho(p_i.\ell) = \rho(p_j.\ell)|$ returns 1 if locations $p_i.\ell, p_j.\ell$ reside in the same region.

Next, we define the *diversity* of a summary.

Definition 4. *The* diversity $div(S)$ *of a summary* S *captures the degree to which the posts in* S *carry dissimilar information. As before, diversity is defined as the weighted sum of a textual and spatial term:*

$$div(S) = \alpha \cdot div^T(S) + (1 - \alpha) \cdot div^S(S) \tag{6}$$

□

Textual diversity is defined with respect to the vector space model. Specifically, textual diversity is the sum of cosine distance between all pairs of posts in the summary:

$$div^T(S) = \sum_{\{p, p'\}: p \neq p' \in S} \left(1 - \sum_{\psi} p_i[\psi] \cdot p_j[\psi]\right) \tag{7}$$

On the other hand, *spatial diversity* is defined based on a spatial distance (e.g., Euclidean, haversine) between summary posts' exact locations:

$$div^S(S) = \sum_{\{p, p'\}: p \neq p' \in S} dist(p.\ell, p'.\ell) \tag{8}$$

Based on the definitions of these two quality measures of a summary, we are now ready to state our problem.

Problem 1. *For each sliding window* $\mathcal{W}$ *over a stream of posts, determine the summary* S^* *of size* k *that maximizes the objective function:*

$$S^* = \underset{S \subseteq \mathcal{W}, |S| = k}{\arg\max} f(S),$$
$$f(S) = \lambda \cdot cov(S) + (1 - \lambda) \cdot div(S)$$

where λ *is a weight parameter for coverage and diversity.* □

3 SUMMARY CONSTRUCTION

If we consider any individual instantiation of the sliding window, our problem formulation is identical to the max-sum diversification problem [5]. Thus, one can apply the adaptation of the greedy algorithm in [1] to summarize the contents of each window. However, such an approach is impractical because the sliding window can be arbitrarily large, thus storing its entire contents is not an option. Instead, we need to devise an efficient solution that operates on limited memory.

To achieve this we need to address two tasks. The first is *how to compute the coverage* of posts without storing the window's full contents. Recall that the coverage of a single post is computed as the sum of its cosine similarity with each post in the window. The second task is *how to construct the summary*, again without having the window's full contents. We address each of these tasks next. Note that, while this problem has been studied for landmark windows with limited memory [9] and for sliding windows without memory restrictions [4], to the best of our knowledge it has not been addressed for sliding windows under limited memory.

3.1 Computing Coverage

To compute the coverage without keeping the entire window contents, we exploit the linearity of the inner product (the cosine similarity of two normalized vectors is their inner product). In what follows, we use the term coverage to refer both to textual and spatial coverage, as they are both defined as a sum of inner products.

Our approach is based on the notion of window *pane* (or subwindow) [6]. For ease of presentation, we assume that the size of the window ω is a factor of its slide step β, e.g., a window of 24 hours sliding every one hour. The window is thus naturally divided into $m = \omega/\beta$ panes. Each time the window slides, all tuples within the oldest pane expire, while new tuples arrive in the newest pane, termed *current*. We denote as $\mathcal{W}$ the current and as $\mathcal{W}'$ the previous window instantiation. We also denote as $\mathcal{W}^-$ the expired pane of the previous window, and refer to the current pane as $\mathcal{W}^+$, i.e., $\mathcal{W}^- = \mathcal{W}' \setminus \mathcal{W}$ and $\mathcal{W}^+ = \mathcal{W} \setminus \mathcal{W}'$. We enumerate the panes of the window by simply using the notation $\mathcal{W}_1$ through $\mathcal{W}_m$. For each pane $\mathcal{W}_i$, we define its *information content* W_i as the vector:

$$W_i = \sum_{p \in \mathcal{W}_i} p. \tag{9}$$

It is then easy to see that the coverage of a post p can be efficiently computed using the information contents of the m panes:

$$cov(p) = \sum_{i=1}^{m} \sum_{\tau} W_i[\tau] \cdot p[\tau], \tag{10}$$

where τ represents either a keyword or a region. This implies a simple solution to computing the coverage. Instead of requiring the set of all posts within a window, it suffices to store only a few vectors, that is the information content of each pane. When the window slides, we just throw away the information content of the expired pane, and begin aggregating posts in the current pane to form its information content.

3.2 Building the Summary

We first present a *Baseline* (BL), which requires storing the entire contents of the window. Thus, this strategy is impractical, serving only as a benchmark to the quality of the summary. The algorithm for this method is an adaptation of the GA algorithm [1]. BL builds the summary incrementally, starting with an empty set. Then, at each step it inserts the post that maximizes the marginal gain of the objective function. Given a summary S, the marginal gain of a post p is:

$$\phi(p) = \lambda \cdot cov(p) + (1 - \lambda) \cdot div(p, S).$$

Note that GA initializes the summary with a random post because it cannot differentiate among posts when the summary is empty. Since BL can differentiate among posts, it selects instead as the first post the one that has the largest coverage.

Next, we describe the *Online Interchange* (OI) algorithm. This is inspired by [9], where an online algorithm for solving the max-sum diversification problem on an ever increasing stream of posts is presented. The proposed technique solves the problem for a landmark window, which spans from a fixed point in the past until the present. For our purposes, we adapt this algorithm to our problem involving sliding windows, where both the start and the end point of the window slide.

The key idea of OI is to construct the summary of the current window by making incremental changes to the summary of the previous window. Thus, initially the summary is the previous summary excluding any expired posts. Then, each newly arrived post is examined in sequence. If the summary is not yet full, the post is simply inserted. Otherwise, the algorithm identifies the best post to evict from the summary in favor of the current examined post. If such a replacement results in an increase of the objective function, the algorithm applies it.

4 EXPERIMENTAL EVALUATION

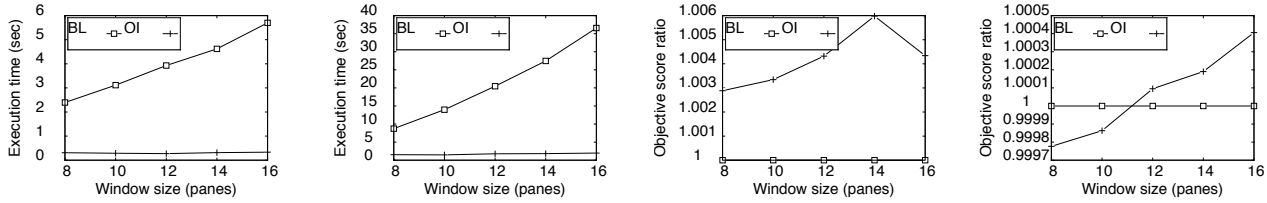
We use two real-world datasets. The first comprises 20 million geotagged images extracted from a publicly available dataset by Yahoo Labs and Flickr[1]. The contained images have worldwide coverage and span a time period of 4 years, from 2010 to 2013. Each image is associated with about 6 keywords on average. The second dataset comprises 20 million geotagged tweets, and is also available online[2]. It has worldwide coverage, and it spans a period of 9 months during 2012. The average number of keywords per post is 5.7.

We process each dataset in a streaming fashion, using the sliding window model, setting the default pane size to $\beta = 4$ hours. We have chosen a rather large value so that the number of posts contained in the resulting panes is in the order of a few thousands, thus essentially compensating for the fact that these datasets are small samples of the actual stream of posts in these sources. The average number of objects per pane is about 2,000 for Flickr and 12,000 for Twitter. Moreover, we set the default window size to $m = 12$ panes, and the default summary size to $k = 15$ objects. Finally, both weight parameters α (Equation 2) and λ (Definition 1) are set to 0.5, thus weighting equally the spatial and textual dimensions, as well as the two criteria of coverage and diversity.

We compare the two methods presented in Section 3.2, namely Baseline (BL) and Online Interchange (OI). To compare their performance, we examine two criteria. Firstly, we investigate their *efficiency*, which is measured as the average execution time required to update the summary every time the window slides. Secondly, we investigate the *quality* of the summaries they produce, by measuring their objective score (see Definition 1). We compute this objective score for each summary a given method produces at every slide of the window, and we take their average over the entire stream. Since computing the optimal summary (i.e., the one that maximizes the objective function) is practically infeasible, we use the objective score achieved by BL as a reference value, and we measure the score of OI as a ratio to that. The algorithms are implemented in Java, and the experiments were conducted on a server with 64 GB memory and an Intel® Xeon® CPU E5-2640 v4 @ 2.40GHz processor, running Debian GNU/Linux 9.0.

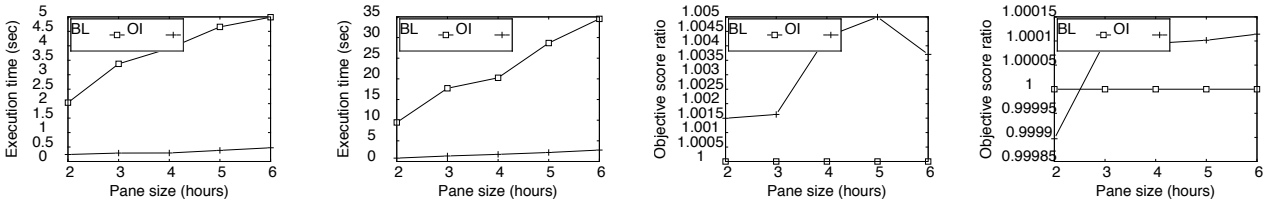
We first examine the execution time of the two methods, varying: (a) the size of the window (number m of panes it contains); (b) the size of each pane (duration β); and (c) the size of each summary (number k of objects). The respective results are shown in Figures 1(a), 2(a) and 3(a) for Flickr and 1(b), 2(b) and 3(b) for Twitter. Notice that, in these plots, logarithmic scale is used on the y axis.

[1] https://code.flickr.net/category/geo/
[2] http://www.ntu.edu.sg/home/gaocong/datacode.htm



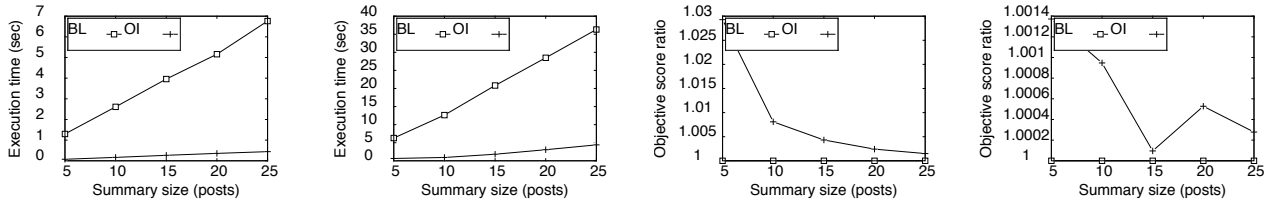
(a) Execution time — Flickr (b) Execution time — Twitter (c) Summary quality — Flickr (d) Summary quality — Twitter

Figure 1: Results for window size (m).



(a) Execution time — Flickr (b) Execution time — Twitter (c) Summary quality — Flickr (d) Summary quality — Twitter

Figure 2: Results for pane size (β).



(a) Execution time — Flickr (b) Execution time — Twitter (c) Summary quality — Flickr (d) Summary quality — Twitter

Figure 3: Results for pane size (k).

As expected, OI outperforms BL in all cases. This is because, in BL, the previous summary is discarded and the new one is computed from scratch, taking into account all posts in the window. Instead, OI constructs the new summary incrementally, discarding only the expired posts from the previous one, and considering only the newly arrived posts as candidates.

Next, we investigate the objective score achieved by the summaries computed by each method. As explained above, we use the objective score of BL as reference. We examine how the results vary for different values of the window size m, pane duration β, and summary size k. The results are shown in Figures 1(c), 2(c) and 3(c) for Flickr and 1(d), 2(d) and 3(d) for Twitter. Interestingly, OI appears to surpass the score of BL. Yet, the observed differences are rather marginal, not exceeding 1%. Essentially, this indicates that OI not only does not suffer a penalty for not rebuilding the summary from scratch, but instead the summary produced by interchanging posts on this basis has a similar, or even slightly higher, quality than the former. Subsequently, OI is clearly the best choice overall considering both performance measures.

ACKNOWLEDGEMENTS

This work was partially supported by the EU H2020 Project City.Risks (H2020-FCT-10-2014-653747).

REFERENCES

[1] B. E. Birnbaum and K. J. Goldman. An improved analysis for a greedy remote-clique algorithm using factor-revealing lps. In *APPROX-RANDOM*, volume 4110 of *Lecture Notes in Computer Science*, pages 49–60. Springer, 2006.

[2] J. G. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336. ACM, 1998.

[3] M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Record*, 39(1):41–47, 2010.

[4] M. Drosou and E. Pitoura. Diverse set selection over dynamic data. *IEEE Trans. Knowl. Data Eng.*, 26(5):1102–1116, 2014.

[5] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.

[6] J. Li, D. Maier, K. Tufte, V. Papadimos, and P. A. Tucker. Semantics and evaluation techniques for window aggregates in data streams. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 311–322, New York, NY, USA, 2005. ACM.

[7] H. Lin and J. A. Bilmes. A class of submodular functions for document summarization. In *ACL*, pages 510–520, 2011.

[8] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[9] E. Minack, W. Siberski, and W. Nejdl. Incremental diversification for very large sets: a streaming-based approach. In *SIGIR*, pages 585–594, 2011.

[10] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[11] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. T. Jr., and V. J. Tsotras. On query result diversification. In *ICDE*, pages 1163–1174, 2011.